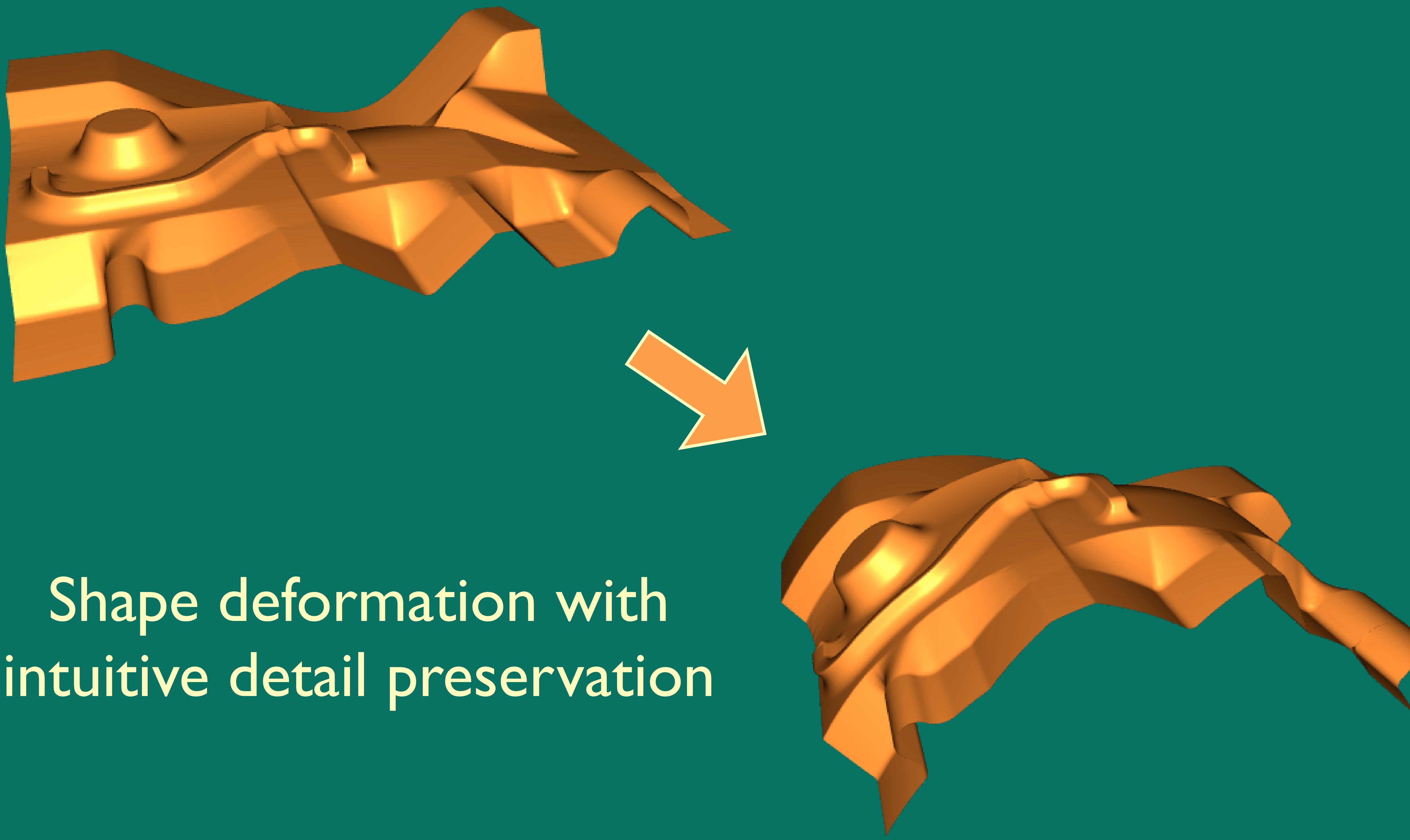
A Remeshing Approach to Multiresolution Modeling

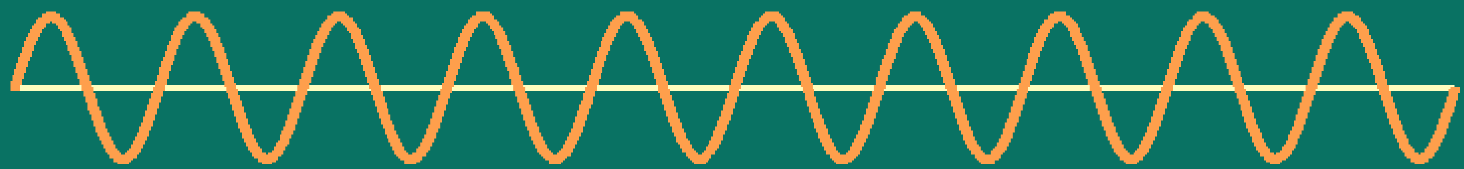
Mario Botsch Leif Kobbelt

Multiresolution Modeling



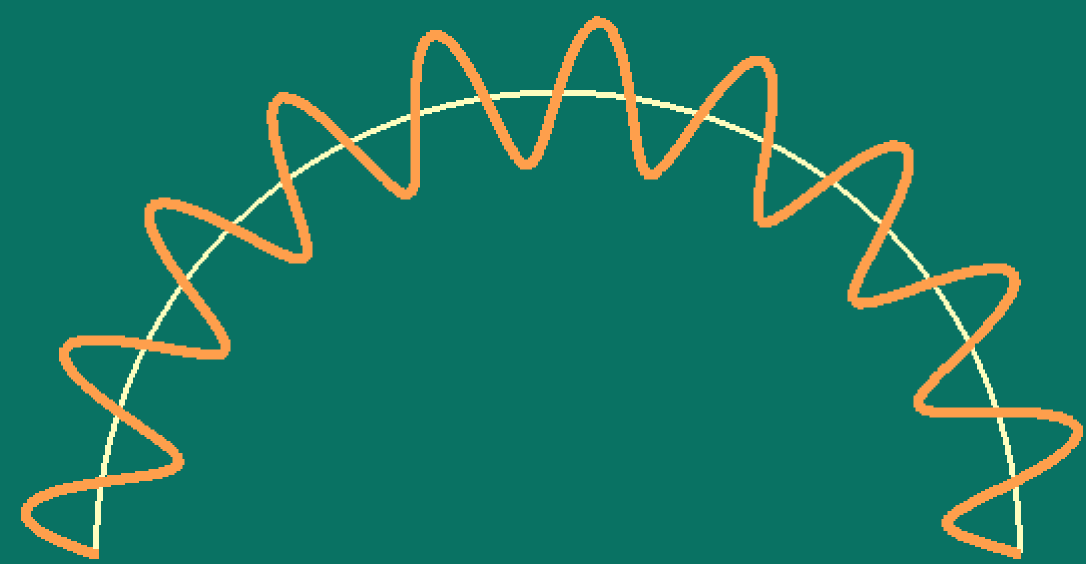
Shape deformation with intuitive detail preservation

Multiresolution Modeling



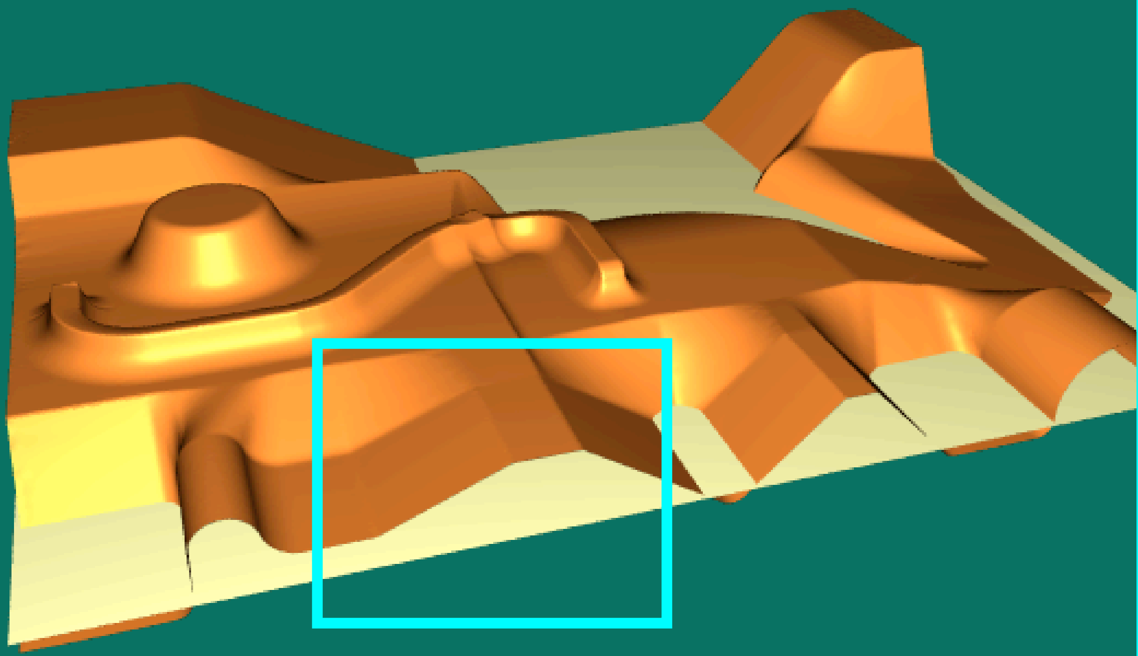
Frequency decomposition

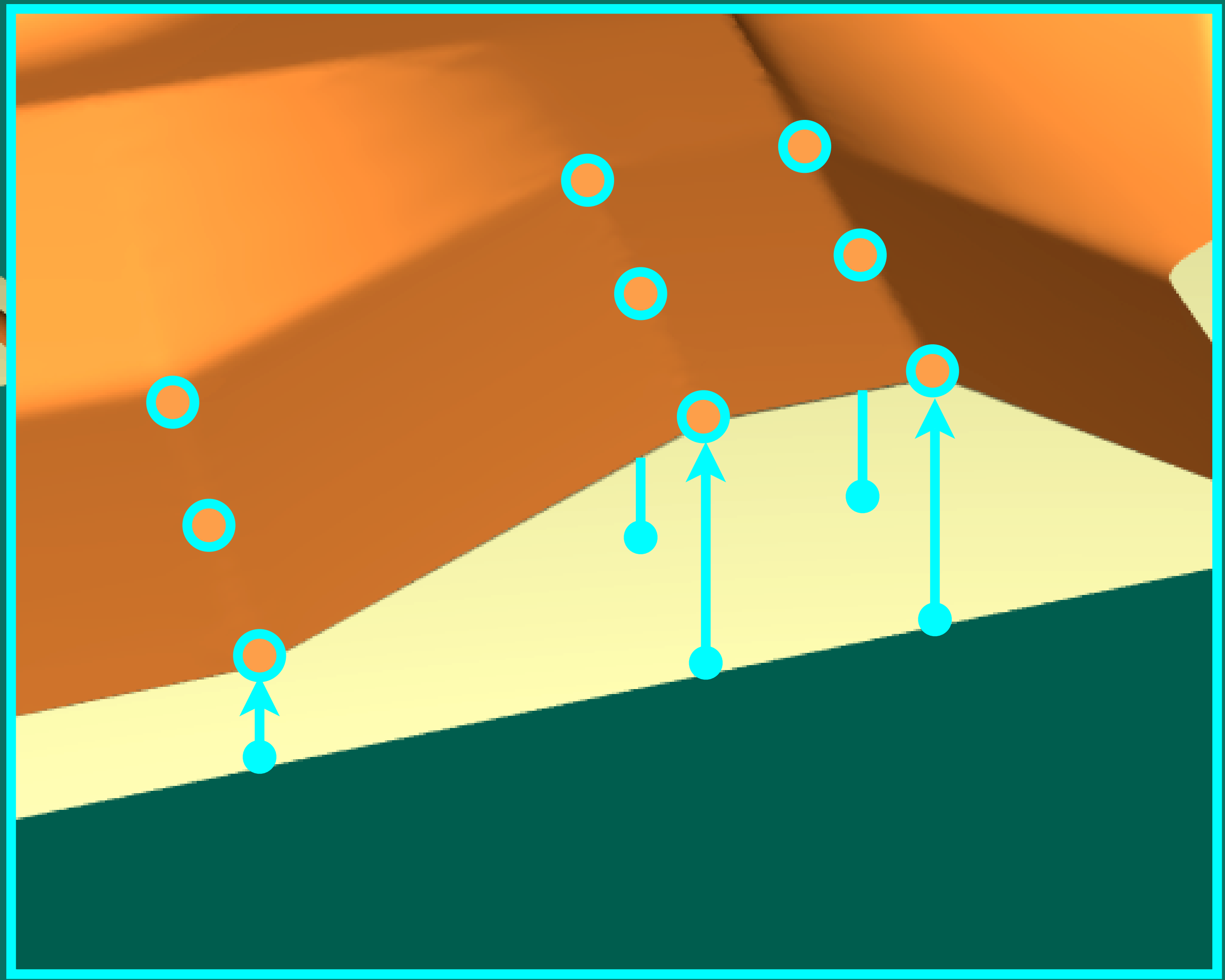
Change low frequencies



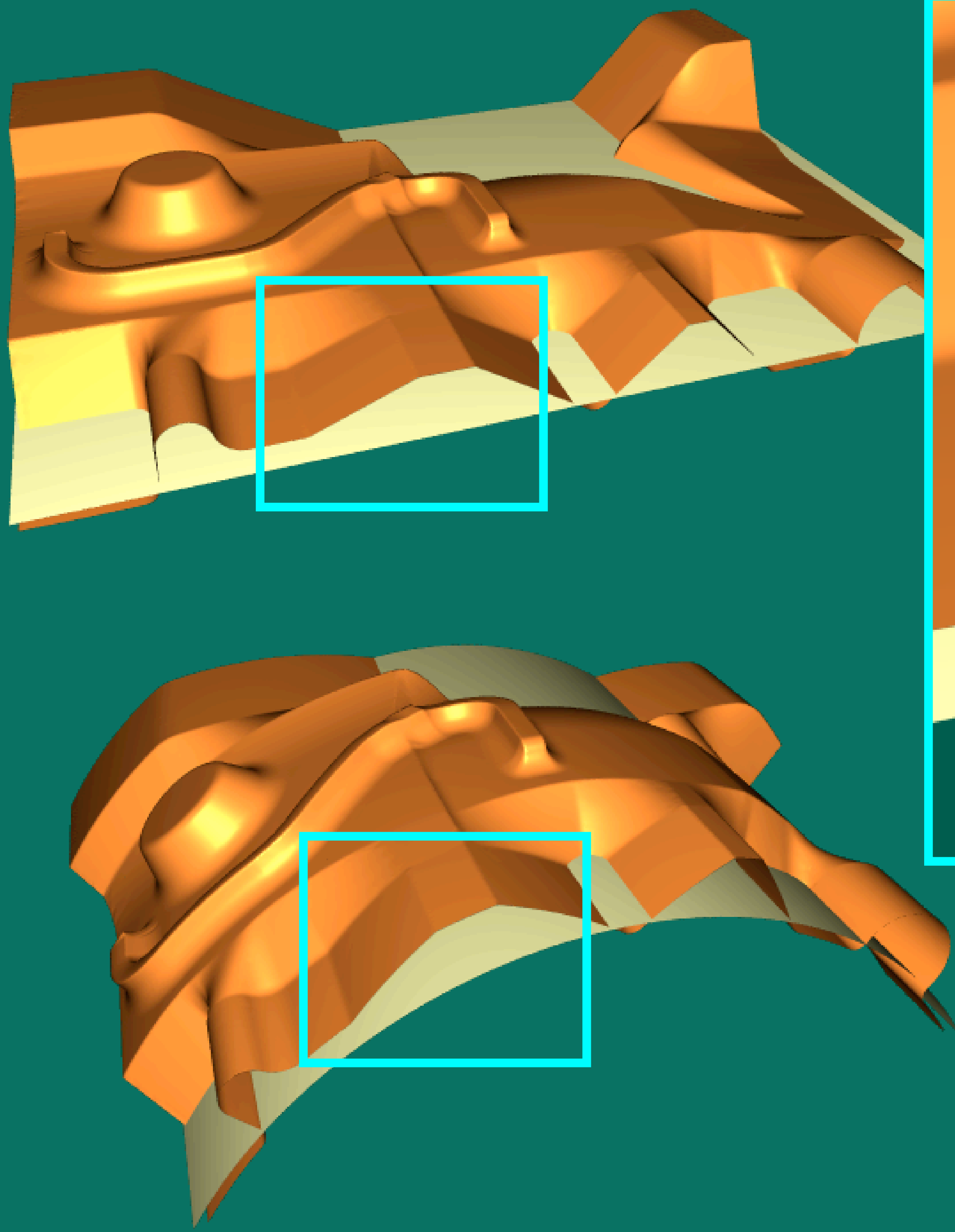
Local frame details

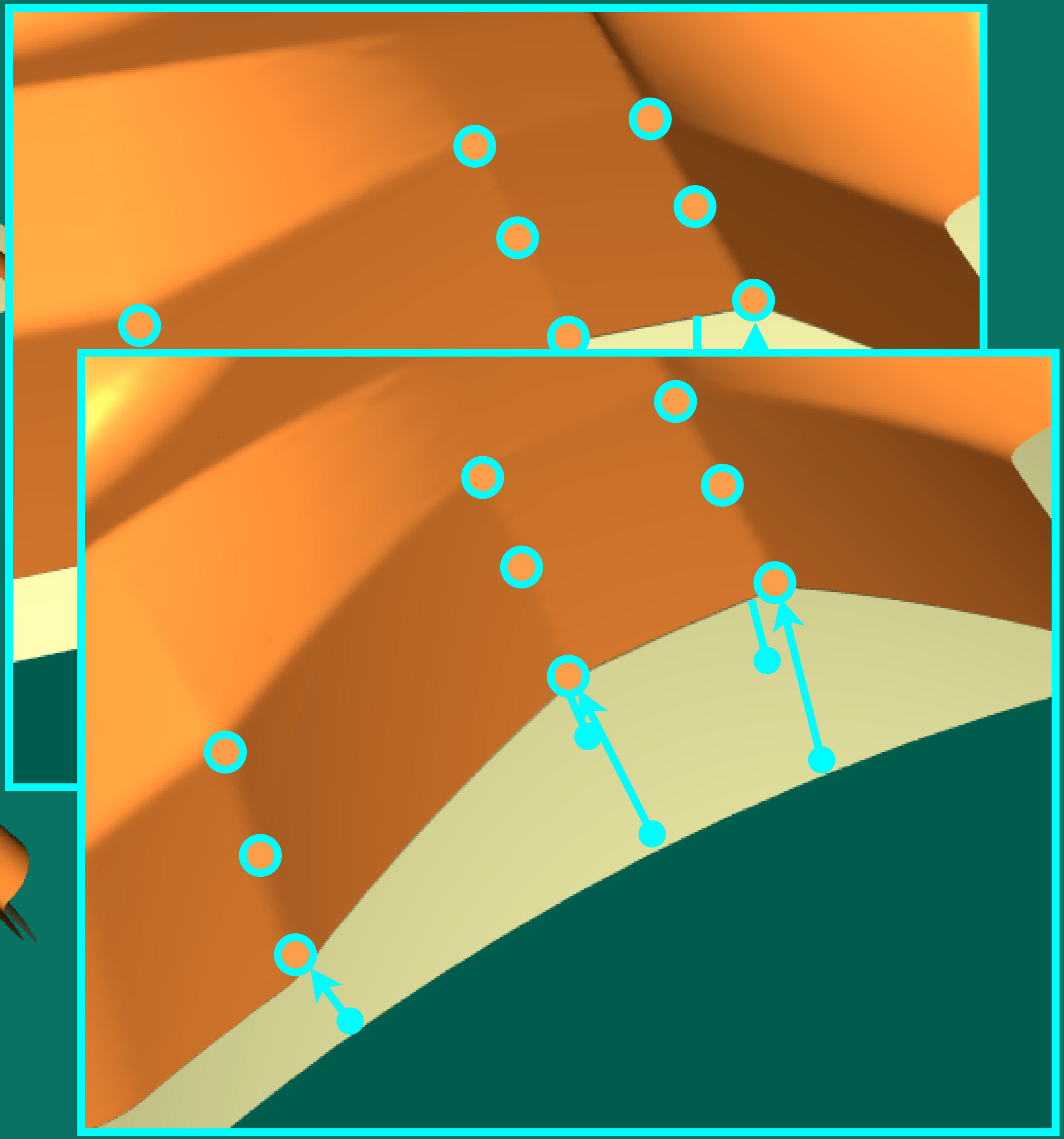
Multiresolution Modeling



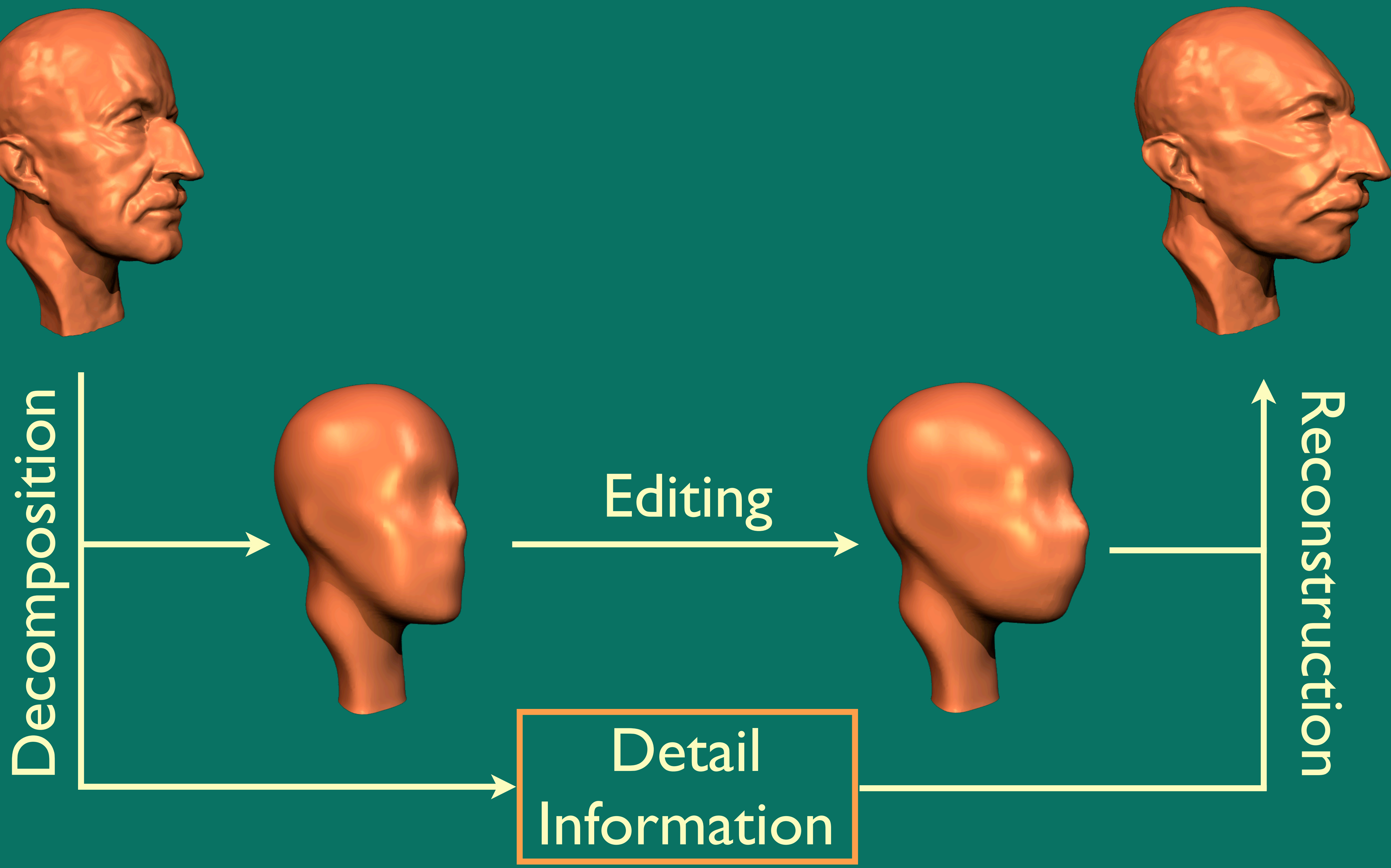


Multiresolution Modeling





Multiresolution Modeling



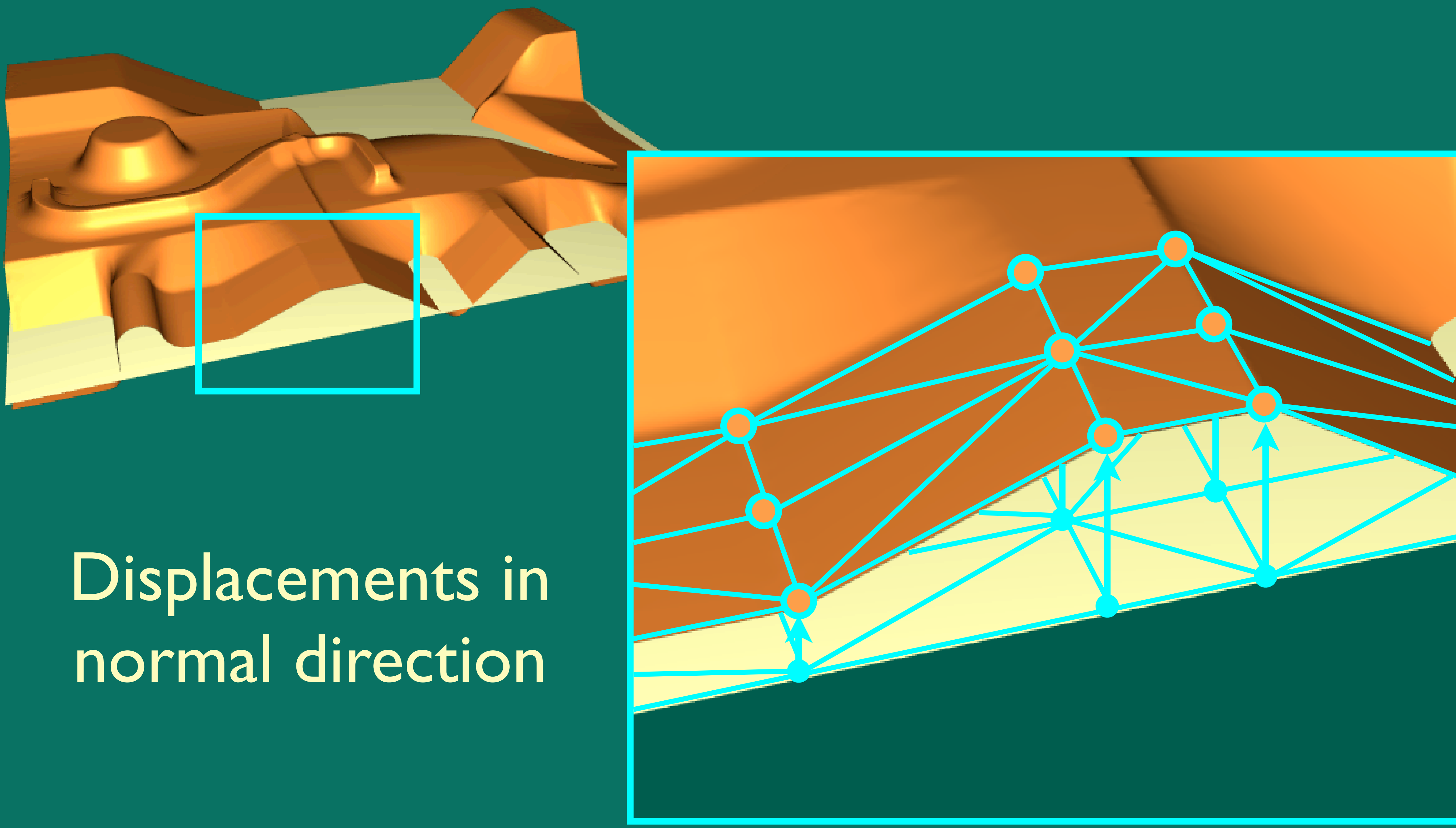
Two Different Meshes

Detailed

Base

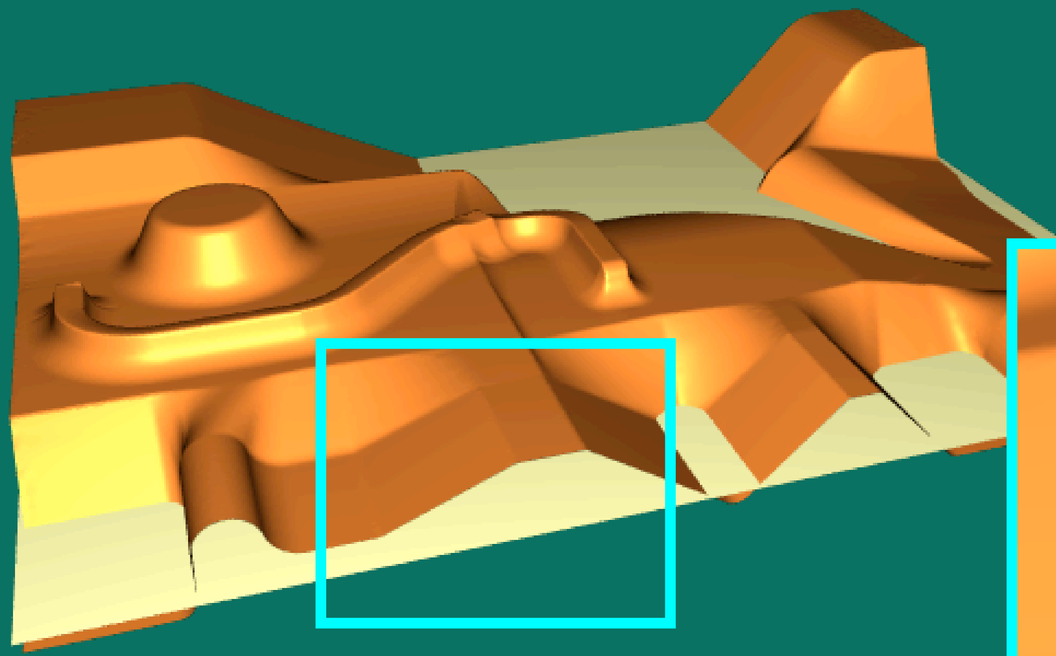
- User interaction
- Decomposition operator
-
-
- Deformation operator
- Reconstruction operator
- Responsible for robustness & efficiency

Detail Encoding

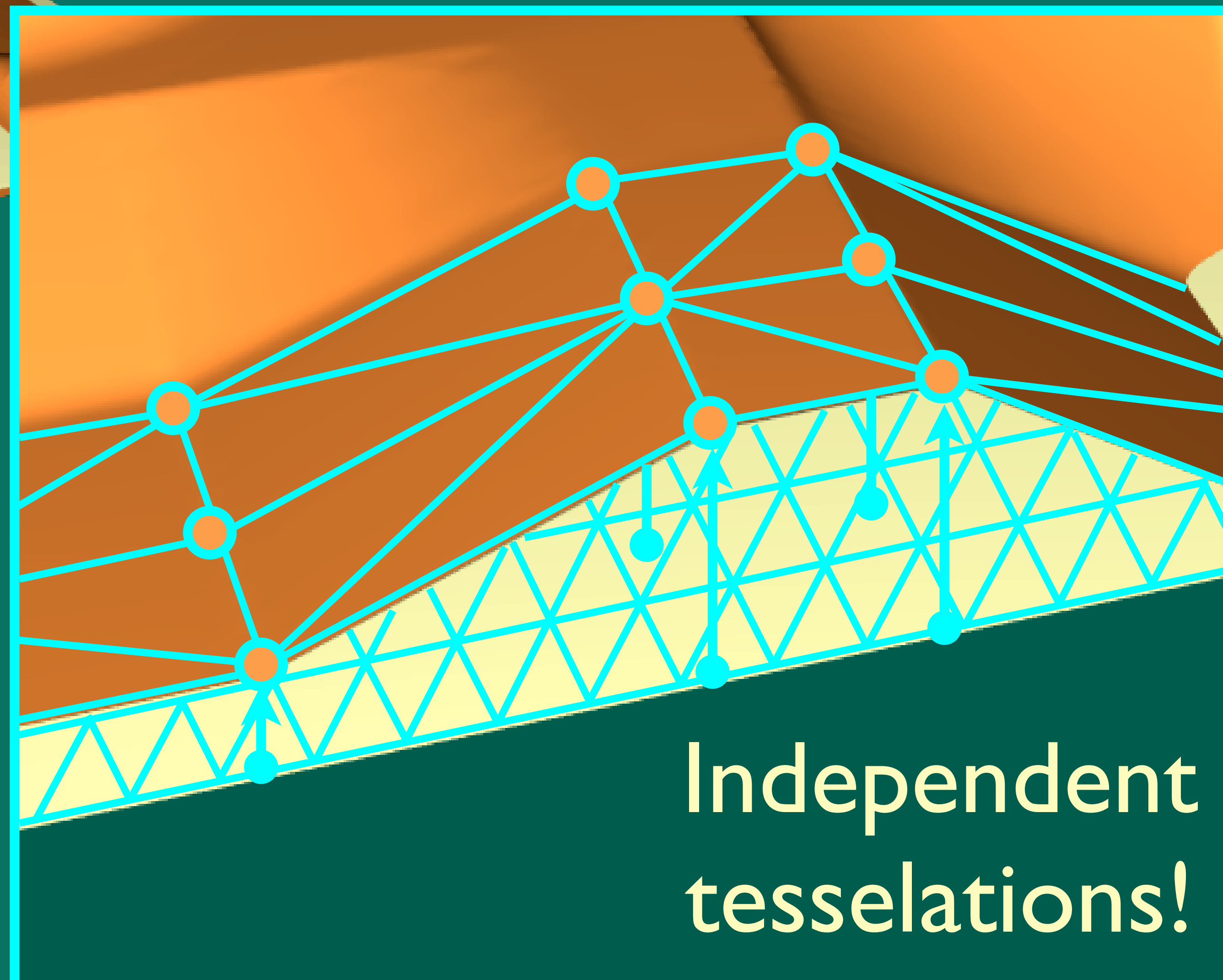


Displacements in normal direction

Detail Encoding



Displacements in normal direction



Remeshing ?

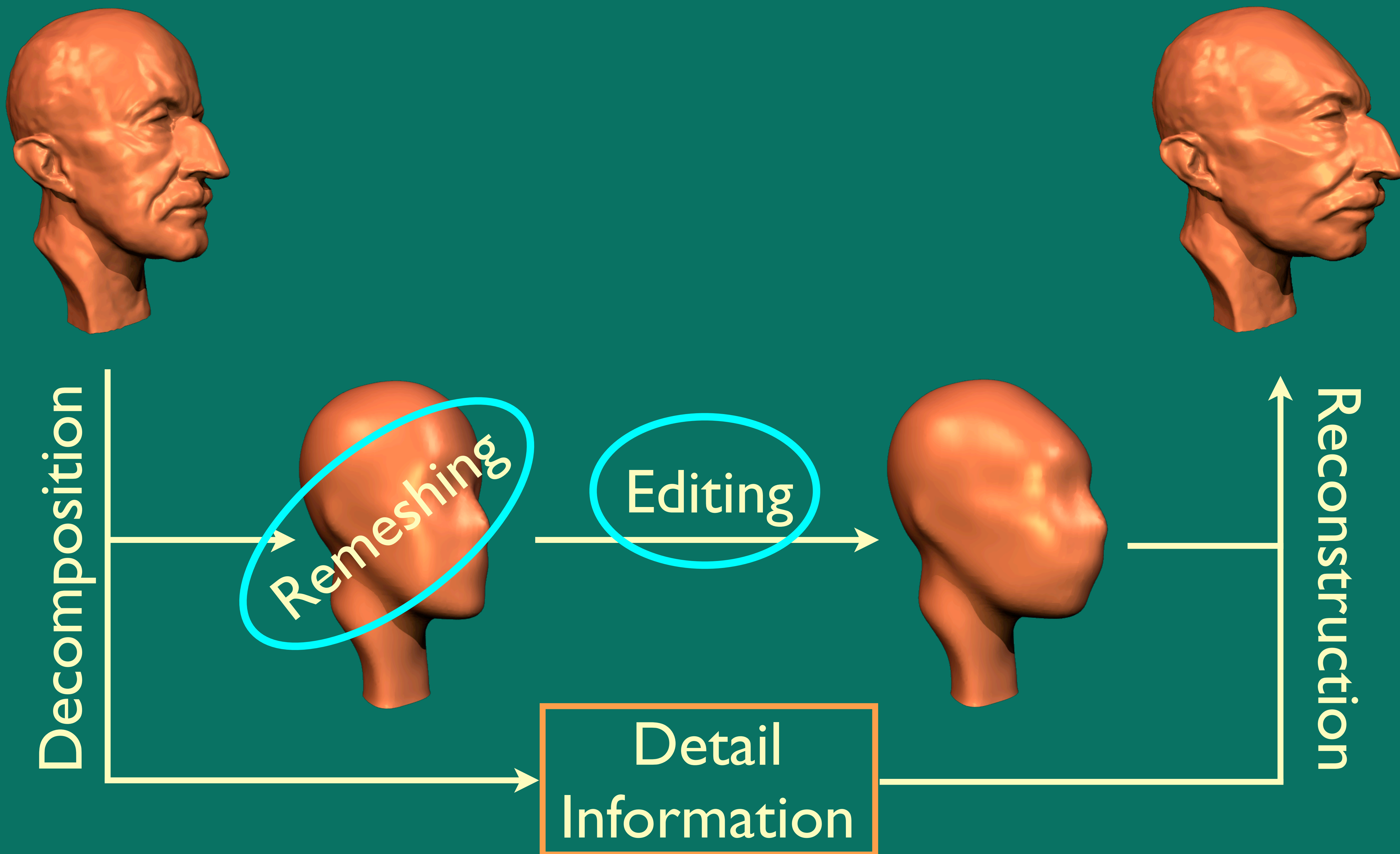
Detailed

Base

- Features, sharp edges
- Hand-crafted triangulation
-
-
-
- Low frequency surface
- No aliasing problems

➡ Remesh base surface

Multiresolution Modeling



Outline

- Introduction
- Freeform Modeling
- Remeshing
- Results

Modeling Requirements

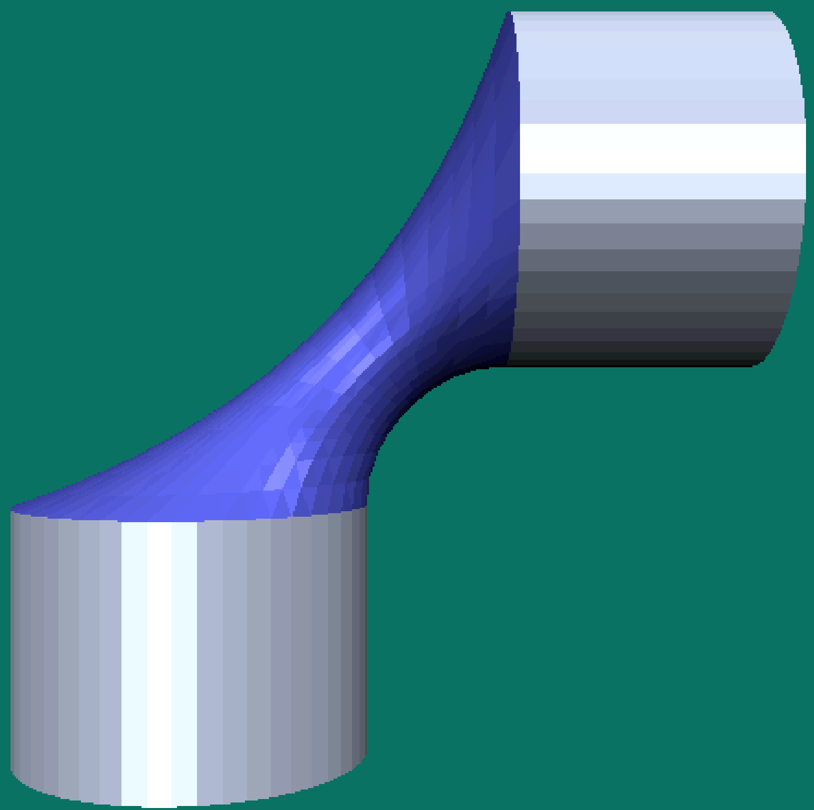
- Per-vertex interpolation constraints
  - Arbitrary support
- Physically plausible behaviour
  - Stiffness, smoothness

Boundary Constraint Modeling

- Prescribe boundary constraints
- vertex positions
- C^0 - C^2 continuities
-
- Constraint energy minimization
- $$E_k(S) = \int F_k\left(S_{u^k}, S_{u^{k-1}v}, \ldots, S_{v^k}\right)$$
- Euler-Lagrange PDE:

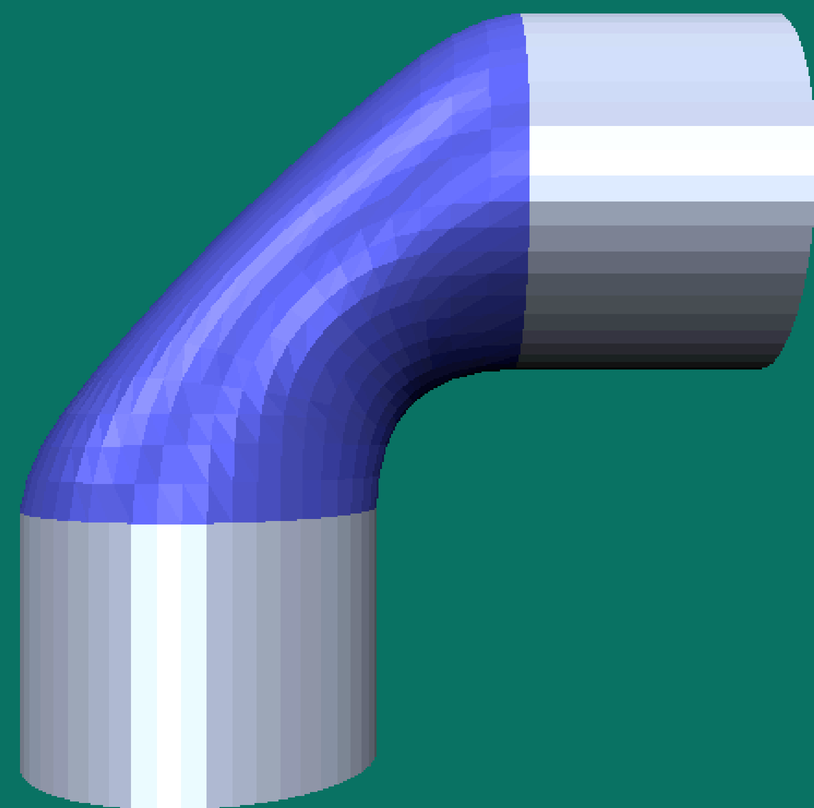
$$\Delta^k(S) = 0$$

Energy Functionals



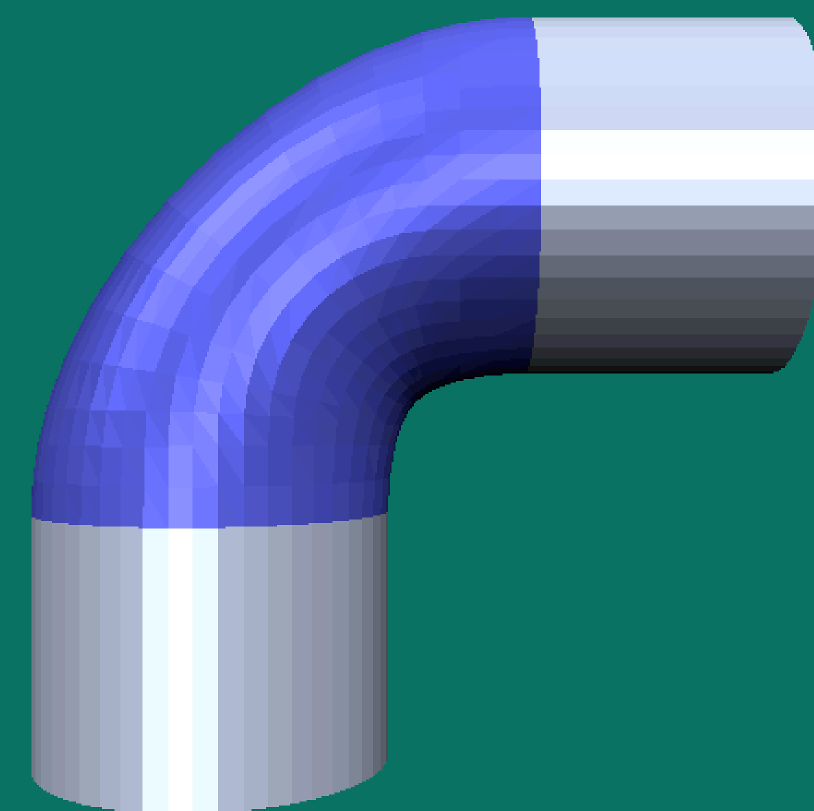
Membrane

$$\Delta S = 0$$



Thin-Plate

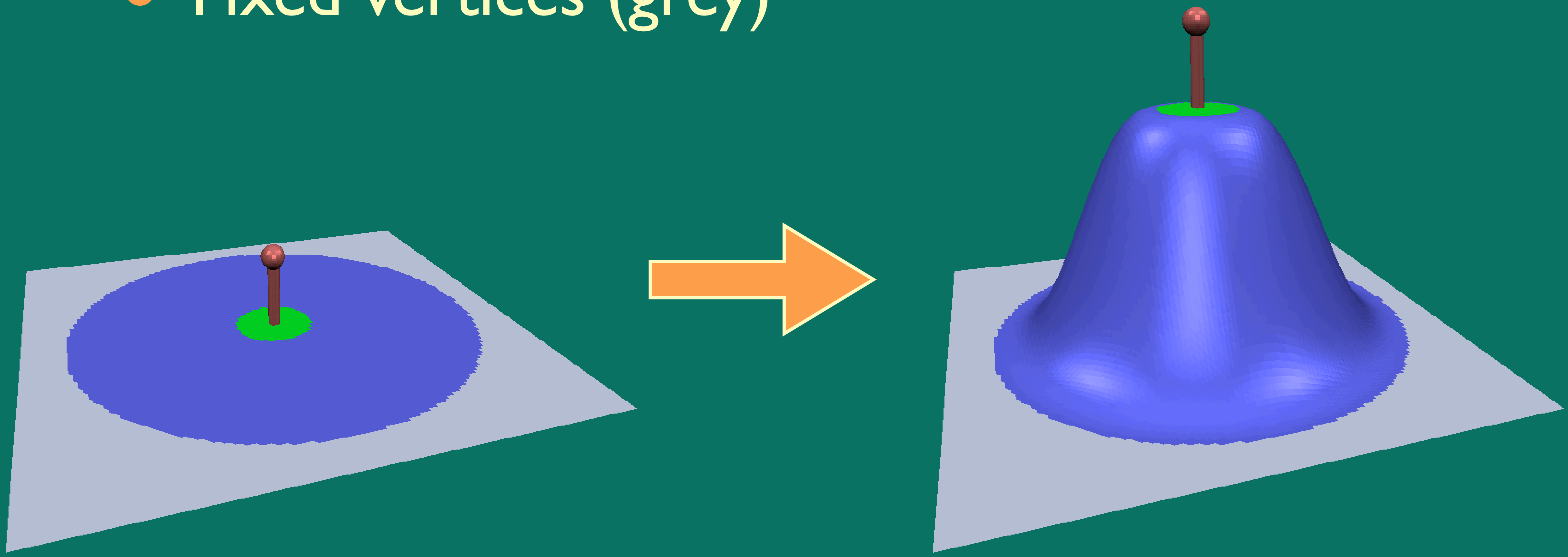
$$\Delta^2 S = 0$$



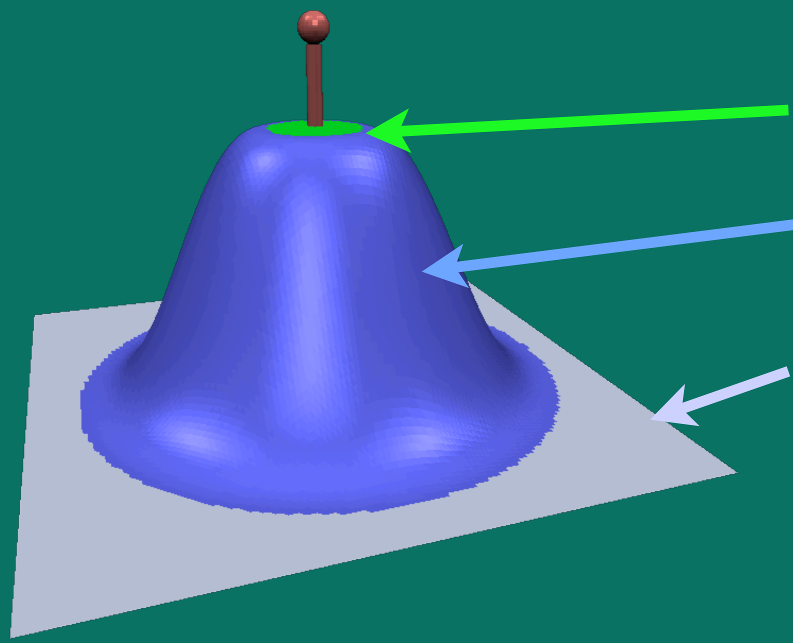
$$\Delta^3 S = 0$$

Modeling Metaphor

- Support region (blue)
- Handle regions (green)
- Fixed vertices (grey)



Discretization → Linear System



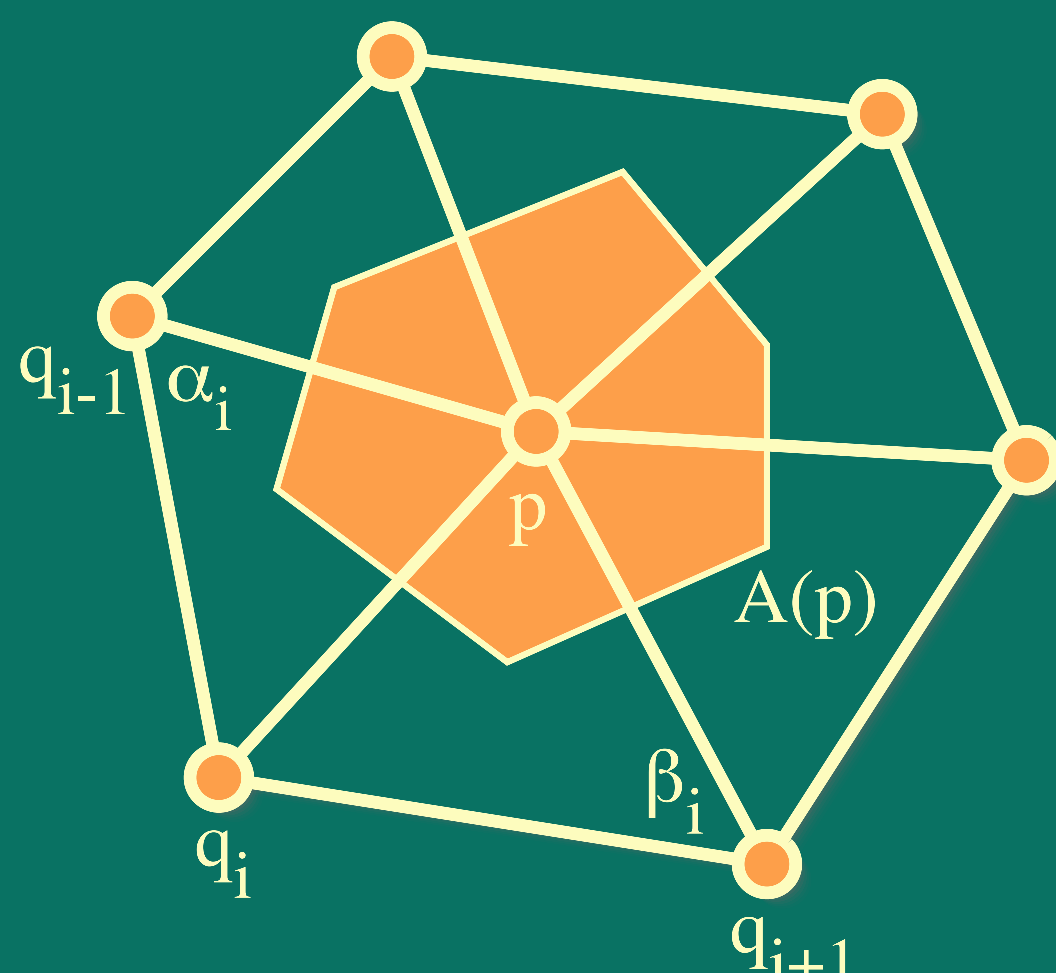
$$\mathbf{h} = \{h_1, \ldots, h_H\}$$
$$\mathbf{p} = \{p_1, \ldots, p_P\}$$
$$\mathbf{f} = \{f_1, \ldots, f_F\}$$

$$\begin{pmatrix} & \Delta^k & \\ \hline 0 & \multicolumn{2}{|c}{I_{F+H}} \end{pmatrix} \begin{pmatrix} \mathbf{p} \\ \mathbf{f} \\ \mathbf{h} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{f} \\ \mathbf{h} \end{pmatrix}$$

$$\Rightarrow \quad \Delta^k \mathbf{p} = \mathbf{b}$$

Laplace Discretization

$$\Delta(p) := \frac{2}{A(p)} \sum_{q_i} (\cot\alpha_i + \cot\beta_i)(p - q_i)$$



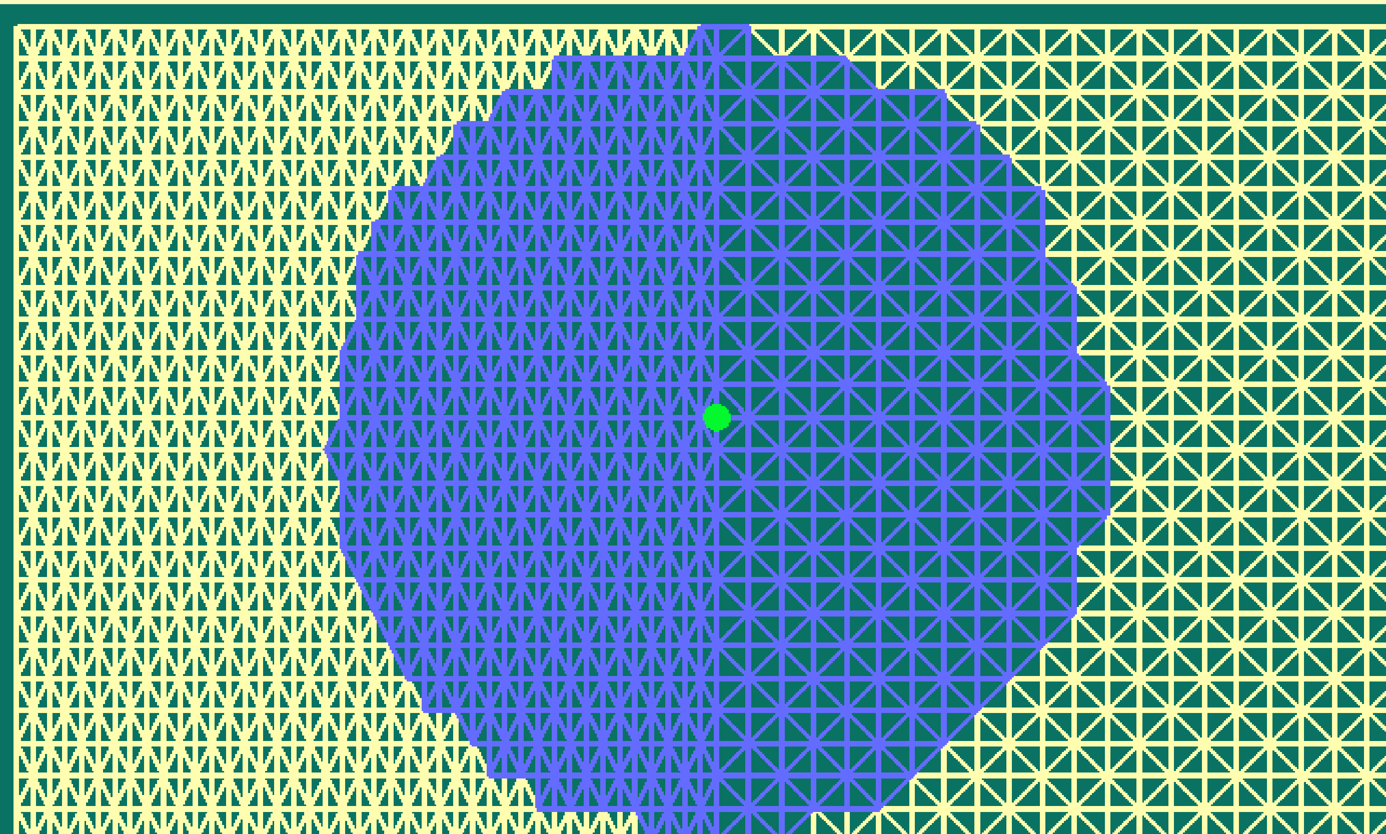
Problems

- Degenerate triangles
  - Matrix no longer positive definite
  - Reconstruction operator unstable

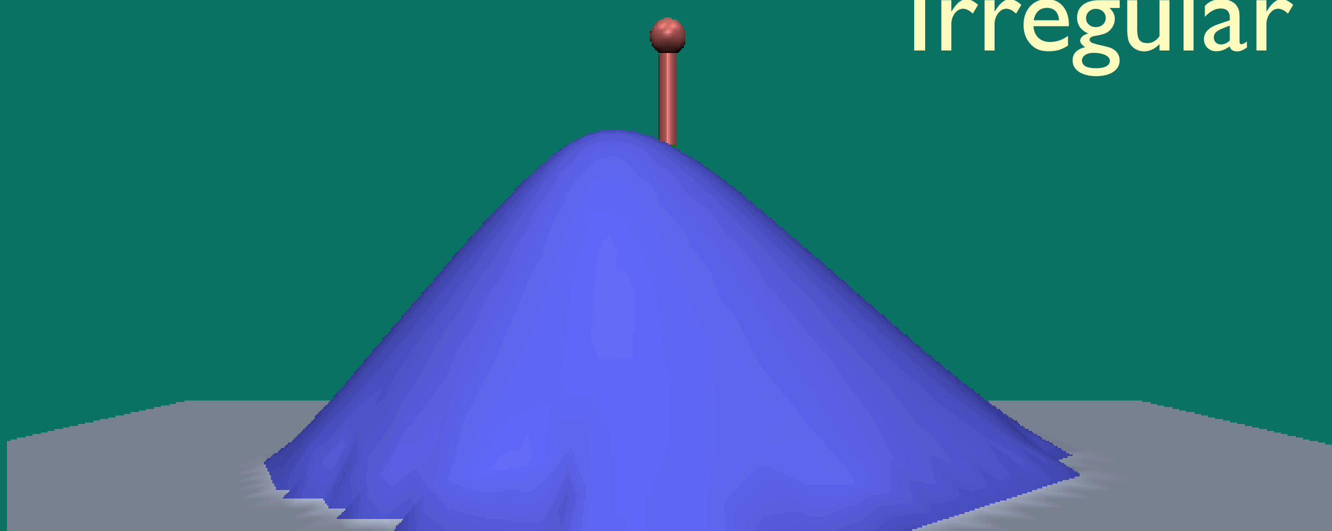
- Matrix unsymmetric
  - Better solvers for symmetric matrices

$$\Delta(p) := \frac{2}{A(p)} \sum_{q_i} (\cot \alpha_i + \cot \beta_i)(p - q_i)$$

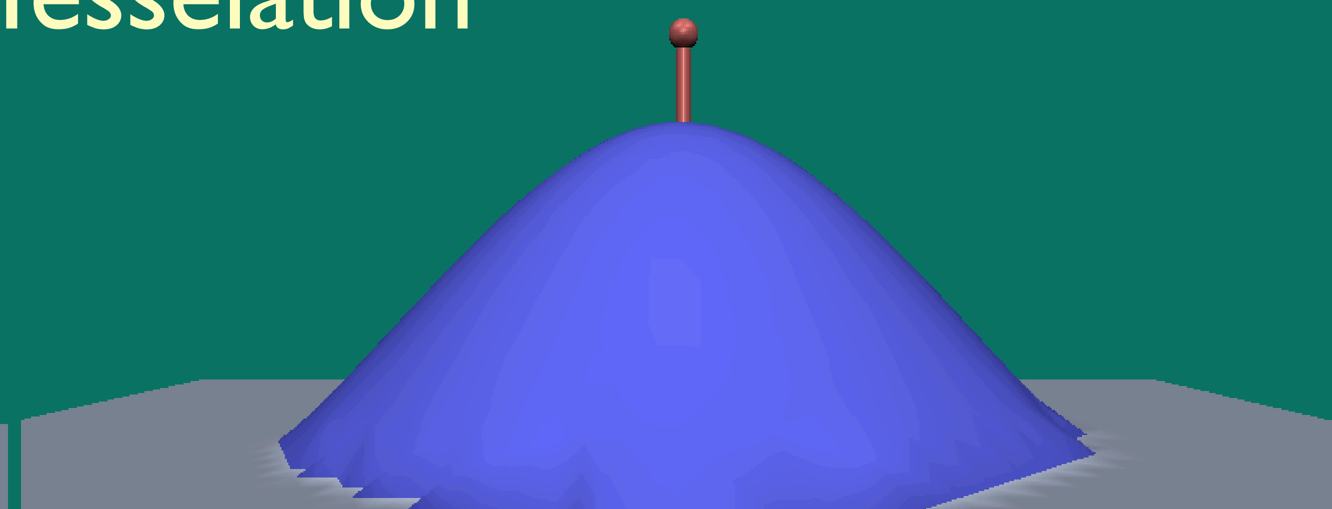
Why not uniform Laplacian?



Irregular Tesselation

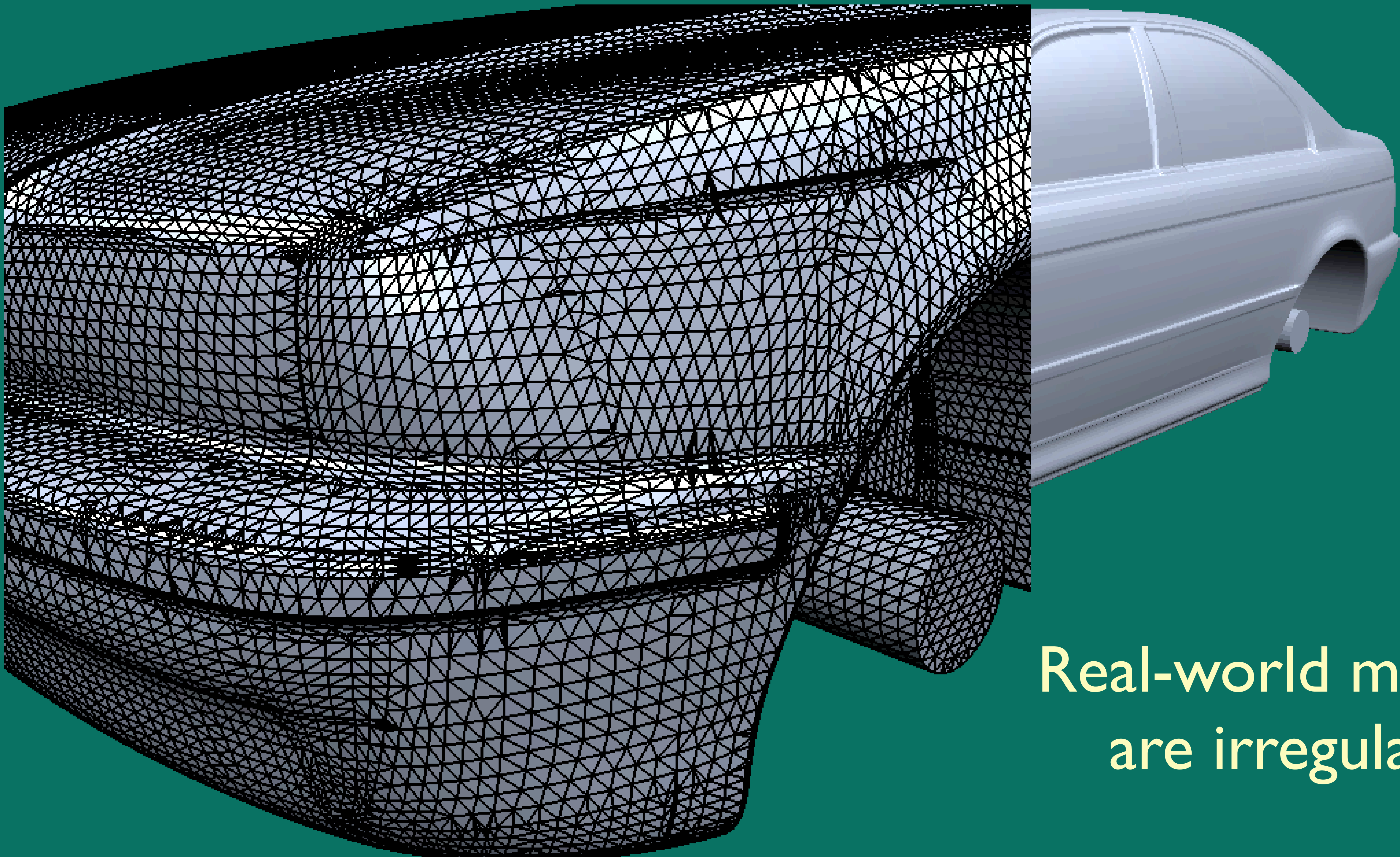


Uniform Weights



Cotangent Weights

Uniform Laplacian Discretization?



Real-world meshes are irregular...

Outline

- Introduction
- Freeform Modeling
- Remeshing
- Results

Remeshing Objectives

- Numerical robustness
  - Triangle roundness
  - <u>Isotropic remeshing</u>
- Computational efficiency
  - Fast linear system solver
  - <u>Symmetric matrix</u>

Isotropic Remeshing

- No global parameterization
  - Explicit remeshing instead
- Several related works:
  - Kobbelt et al. 2000
  - Vorsatz et al. 2003
  - Surazhsky et al. 2003

Isotropic Remeshing

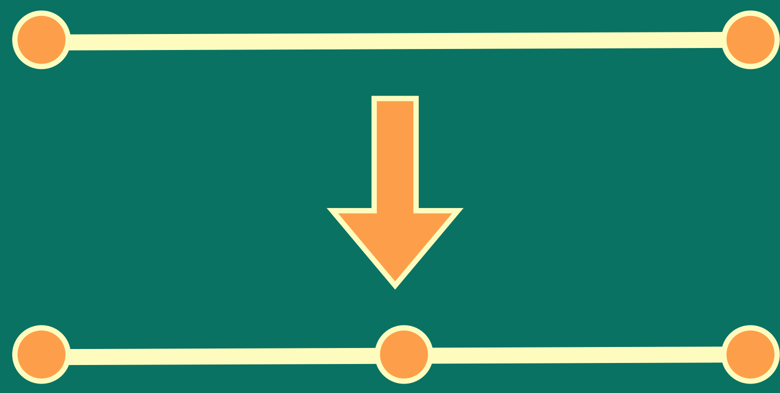
Specify target edge length L

Iterate:

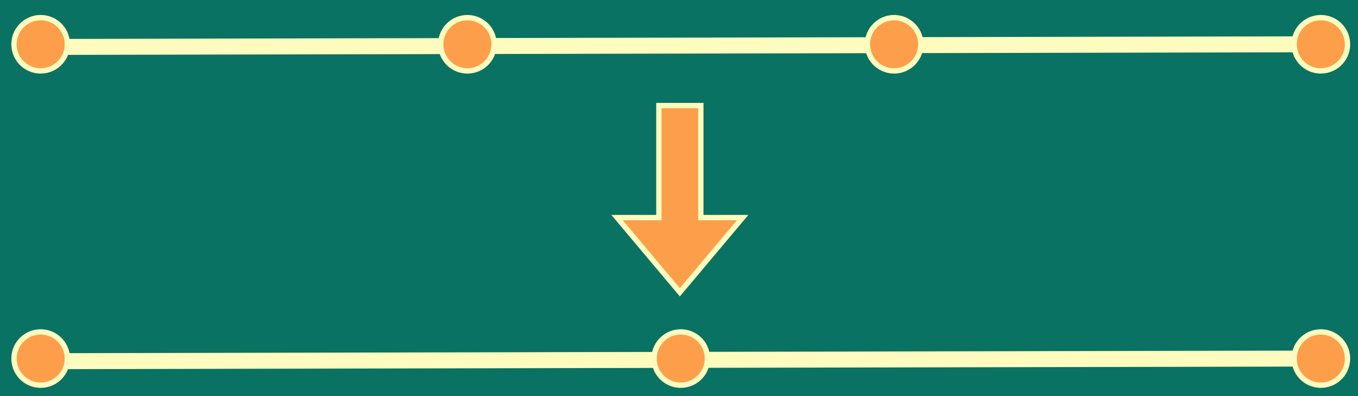
1. Split edges longer than e_{max}
2. Collapse edges shorter than e_{min}
3. Flip edges to get valence 6
4. Relaxation by tangential smoothing

Optimal thresholds?

Edge Length Thresholds

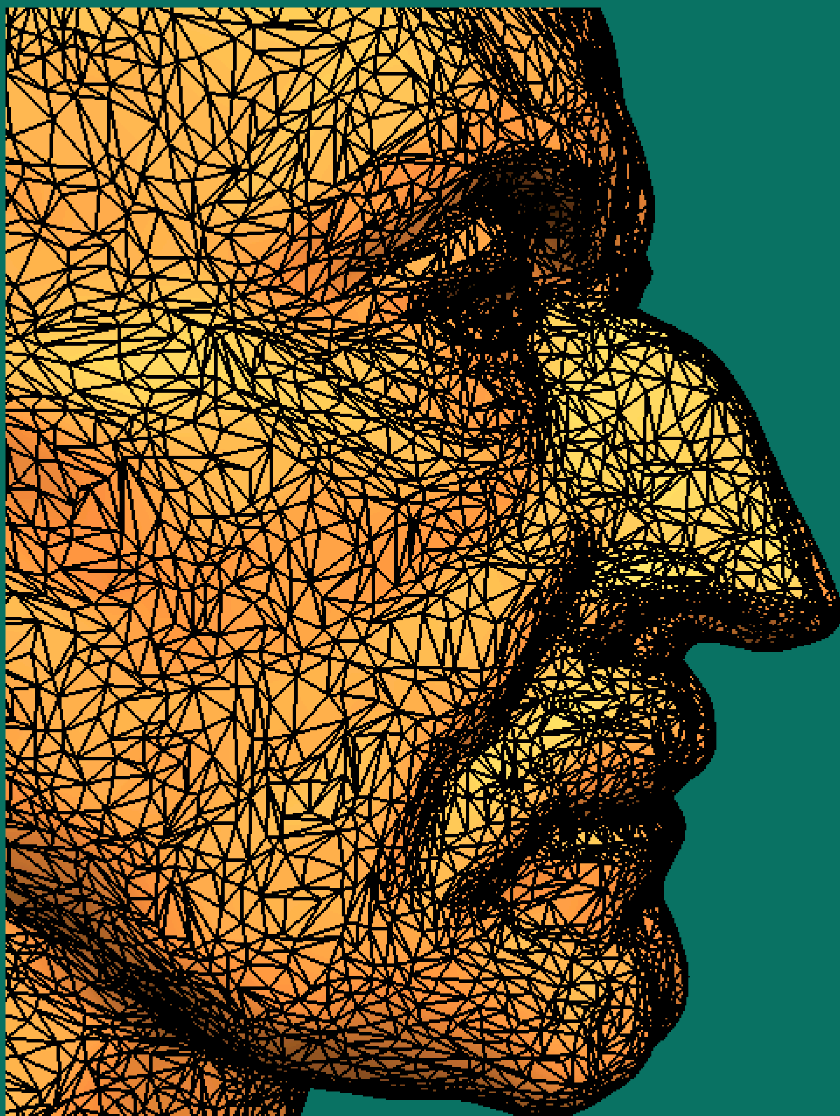


$$|e_{\max} - L| = \left|\tfrac{1}{2}e_{\max} - L\right|$$
$$\Rightarrow e_{\max} = \tfrac{4}{3}L$$

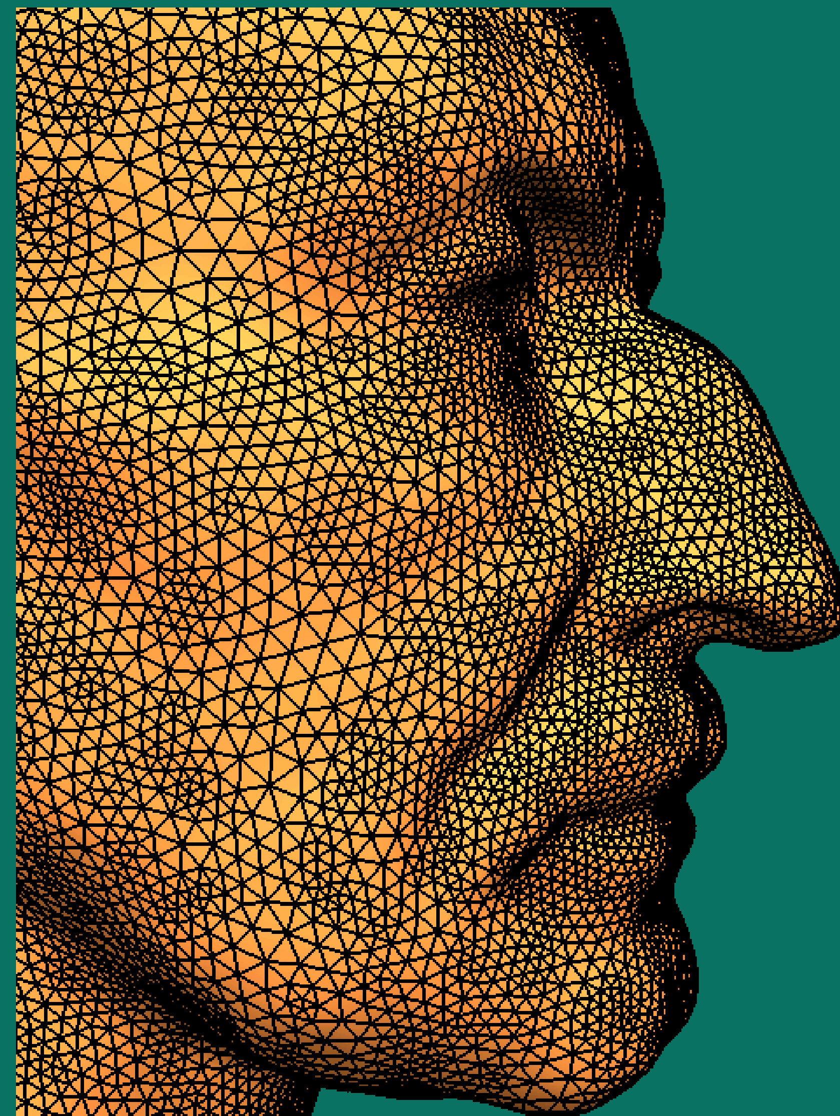


$$|e_{\min} - L| = \left|\tfrac{3}{2}e_{\min} - L\right|$$
$$\Rightarrow e_{\min} = \tfrac{4}{5}L$$

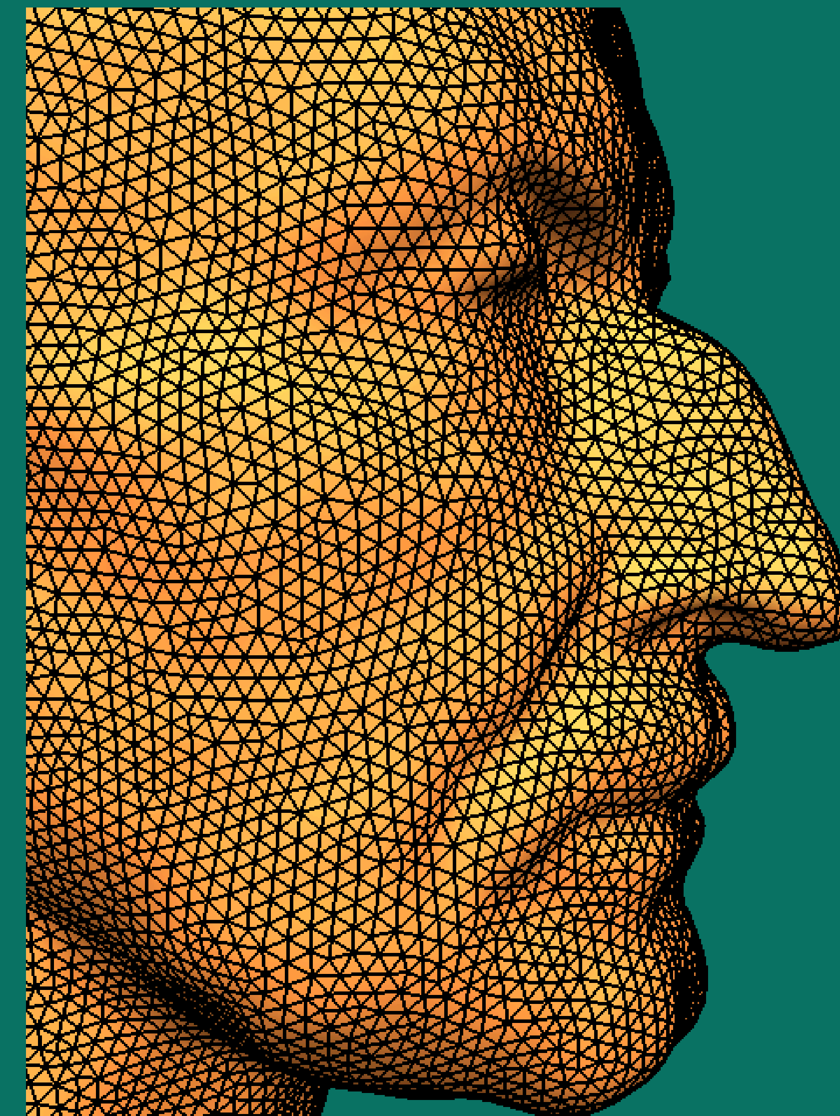
Remeshing Results



Original



$\left(\frac{1}{2}, 2\right)$



$\left(\frac{4}{5}, \frac{4}{3}\right)$

Isotropic Remeshing

- Leads to well-shaped triangles
  - Increased robustness

- But matrix still unsymmetric
  - Because of Voronoi areas A(p)
  - Equalize areas !

$$\Delta(p) := \frac{2}{A(p)} \sum_{q_i} (\cot \alpha_i + \cot \beta_i)(p - q_i)$$

Area Equalization

- Assign mass A(p) to each vertex p
- Mass weighted centroid

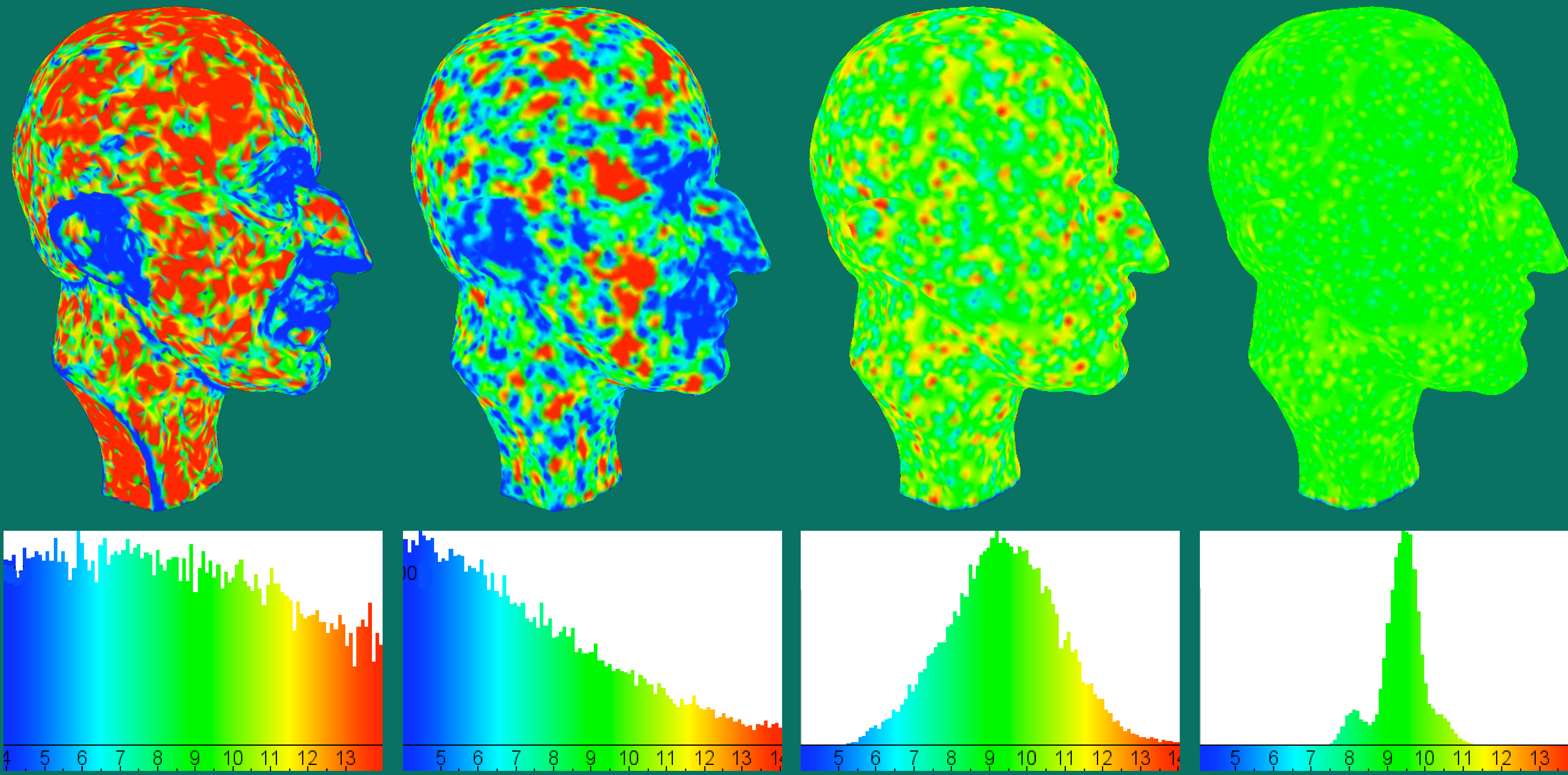
$$\mathbf{g}_i := \frac{1}{\sum_{\mathbf{q}_i} A(\mathbf{q}_i)} \sum_{\mathbf{q}_i} A(\mathbf{q}_i)\, \mathbf{q}_i$$

-
- Tangential update

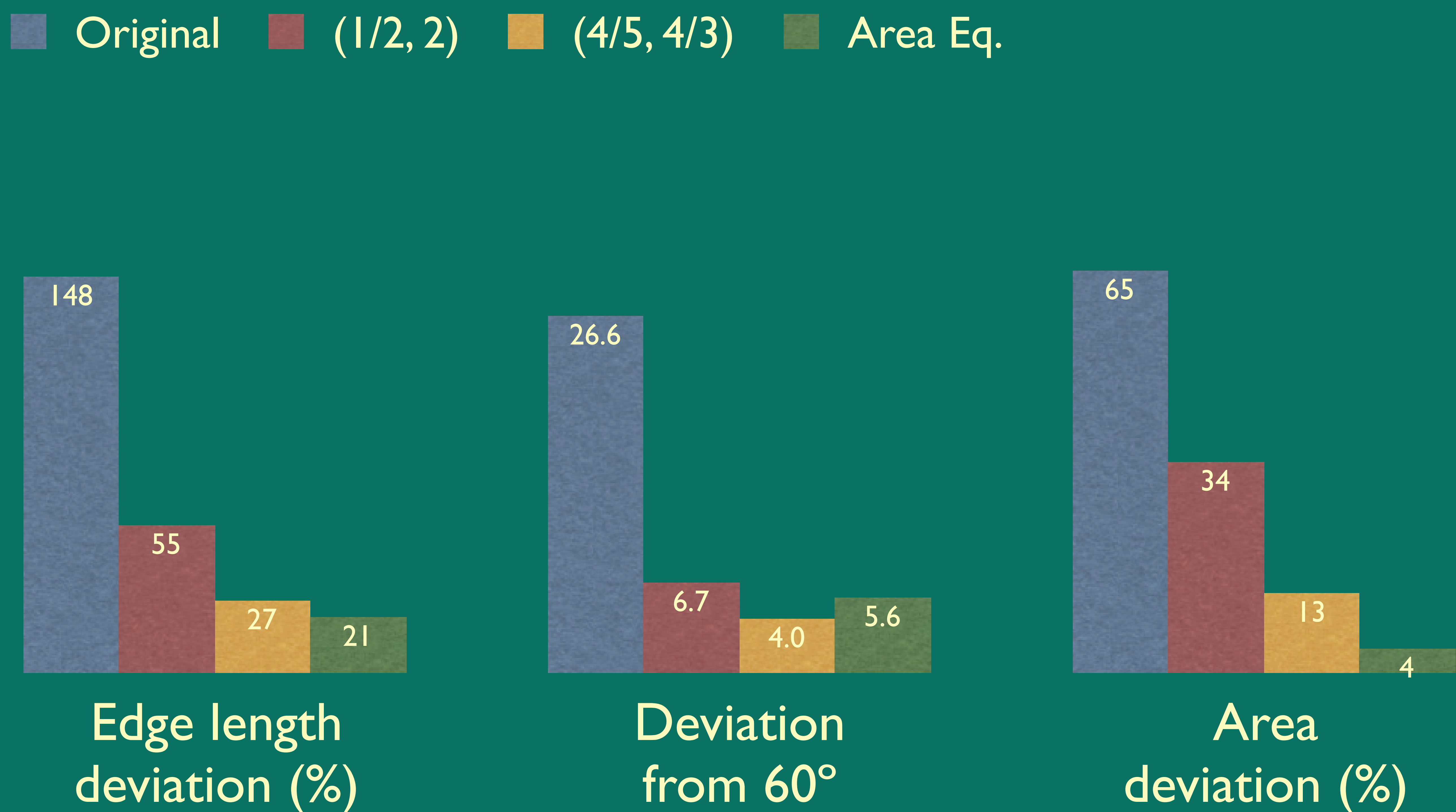
$$\mathbf{p}_i \mapsto \mathbf{p}_i + \lambda \left(I - \mathbf{n}_i \mathbf{n}_i^T \right) (\mathbf{g}_i - \mathbf{p}_i)$$

Remeshing Results

Original | $\left(\frac{1}{2}, 2\right)$ | $\left(\frac{4}{5}, \frac{4}{3}\right)$ | Area Eq.



Remeshing Results



Area Equalization Remeshing

- Efficient algorithm
  - Projection instead of local parametrization
  - Remesh 100k triangles in <5 sec
  -
- Very regular mesh
  - Inner angles close to 60°
  - Relative mean area error <5%

Outline

- Introduction
- Freeform Modeling
- Remeshing
- Results

Increased Robustness

- No degenerate triangles
  - Matrix is positive definite
- No obtuse angles
  - Cotangent weights are positive
  - More reliable Laplacian discretization

Symmetric Laplace Matrix

- Replace Voronoi areas by their mean
- $$\bar{\Delta}(p) := \frac{2}{\bar{A}} \sum_{q_i} (\cot \alpha_i + \cot \beta_i)(p - q_i)$$
-
- Matrix becomes symmetric
- $$\bar{\Delta}^k \mathbf{p} = \mathbf{b}$$
- Small low-frequency errors (~0.7%)
  - Compensated by detail encoding (~0.2%)

Different Solvers

- Iterative solvers
  - Not suitable for large systems: $O(n^2)$
- Multigrid solvers
  - Robust and efficient: $O(n)$
  - Quite complicated to implement
- Direct solvers ?

Direct Solvers

- Naive direct solvers are $O(n^3)$
  - Not suitable for large systems
- System is sparse, not band-limited
  - Band-limitation by reordering
- Band-limited factorizing solvers
  - Factorization: $O(bn^2)$
  - Solving: $O(bn)$

Direct Solvers

- Unsymmetric systems:
  - Band-limited LU factorization
  - Requires pivoting for stability
  - Compromises band-limiting permutations
- Symmetric systems:
  - Band-limited Cholesky factorization
  - Backward stable, exploits symmetry

Comparison

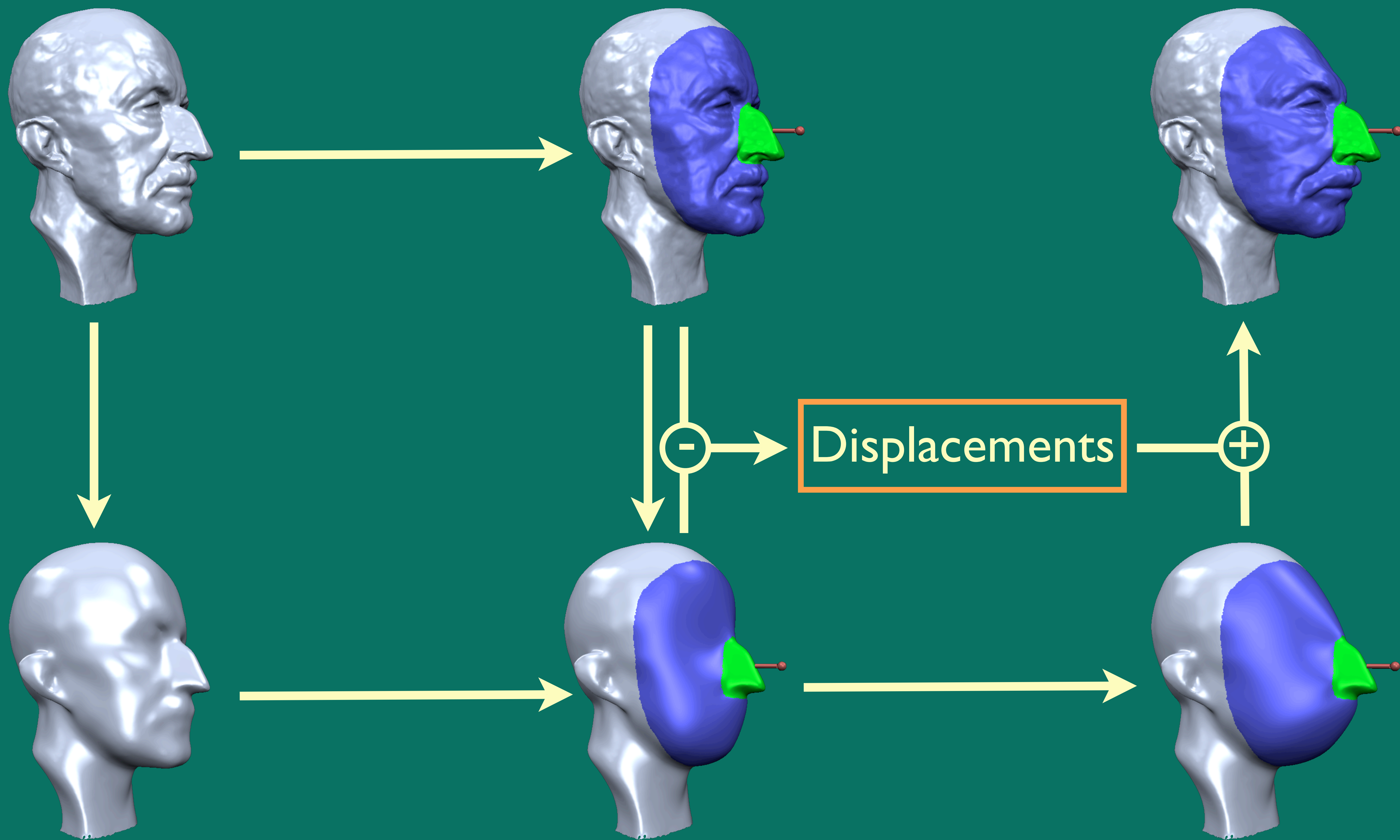
- Iterative solvers
  - Not suitable for large systems: $O(n^2)$
- Multigrid solvers
  - Robust and efficient: $O(n)$
  - Quite complicated to implement
- Direct solvers
  - Same linear complexity
  - Faster by an order of magnitude
  - Considerably easier to use

Comparison (15k DoF)

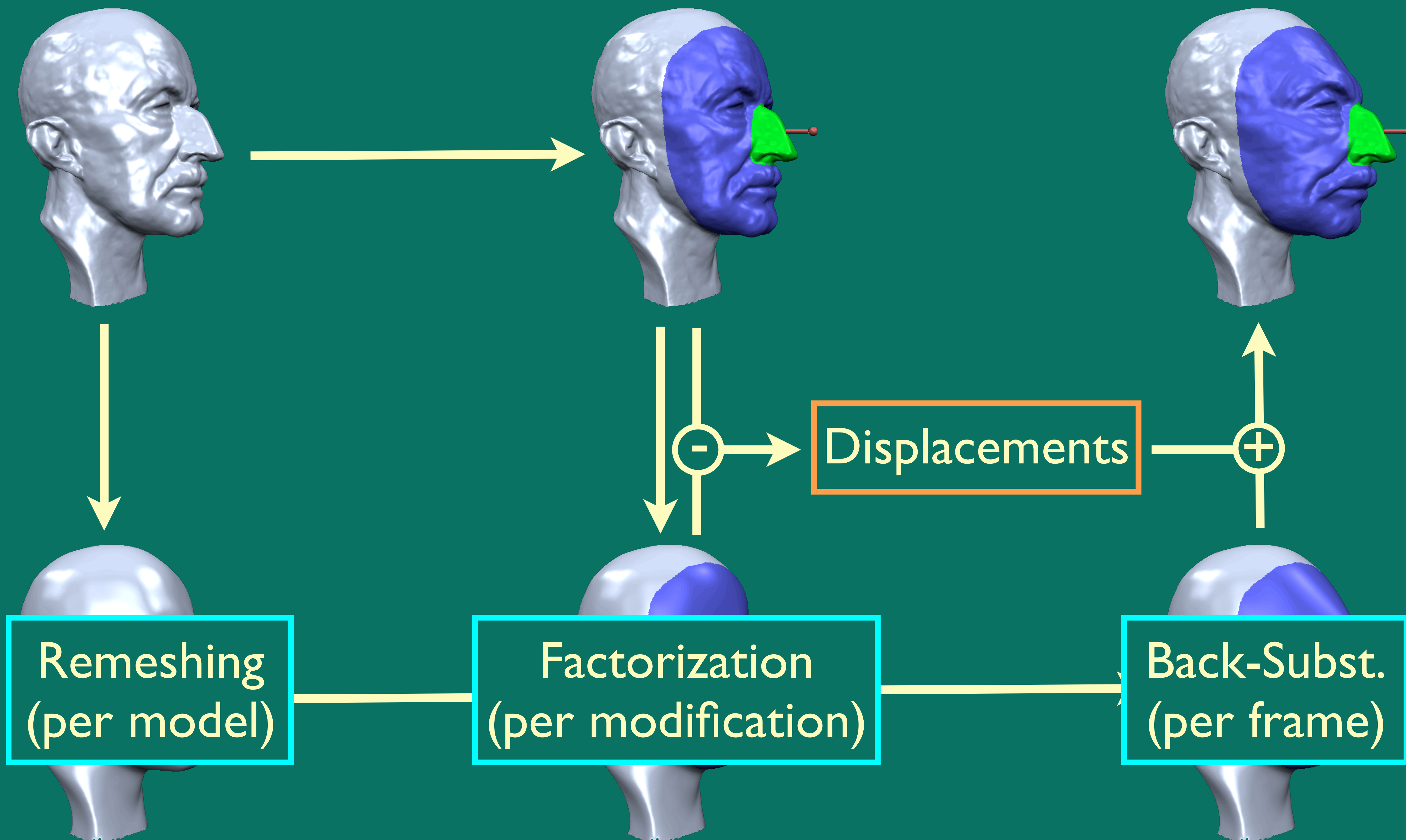
	Precomputation	XYZ Solution	
Iterative	7.2s	7.4s	$O(n^2)$
Multigrid	4.5s	0.8s	$O(n)$
Direct	2.4s	0.07s	$O(n)$

Multigrid vs. Direct

System Overview



System Overview



Conclusion

- Multiresolution framework
  - Independent tesselations
  - Remesh smooth base surface
- Area equalizing isotropic remeshing
  - Improves numerical robustness
  - Yields symmetric matrix
- Allows for direct solvers
  - Significantly faster
  - Easier to use